Evolutionary Fuzzy Markup Language-based Fuzzy Inference Systems with Feature Selection for Winning Rate Prediction in Game of Go

Yuichi Omozaki
Department of Computer Science and Intelligent Systems
Graduate School of Engineering
Osaka Prefecture University
Osaka, Japan
yuichi.omozaki@ci.cs.osakafu-u.ac.jp

Naoki Masuyama
Department of Computer Science and Intelligent System
Graduate School of Enginnering
Osaka Prefecture University
Osaka, Japan
masuyama@cs.osakafu-u.ac.jp

Yusuke Nojima
Department of Computer Science and Intelligent System
Graduate School of Engineering
Osaka Prefecture University
Osaka, Japan
nojima@cs.osakafu-u.ac.jp

Hisao Ishibuchi
Department of Computer Science and Engineering
Southern University of Science and Technology
Shenzhen, China
hisao@sustech.edu.cn

***Abstract*—This paper presents detailed explanations of our technical approach for the FML-based Machine Learning Competition for Human and Smart Machine Co-Learning on Game of Go / AIoT Applications at IEEE WCCI 2020. For this competition, we propose an evolutionary fuzzy markup language-based fuzzy inference system with feature selection for winning rate prediction. Experimental results show that the proposed method can generate a fuzzy inference system with superior prediction performance.**

I. Introduction

The IEEE Computational Intelligence Society has published the first IEEE Standard (IEEE Std 1855-2016) [1] for fuzzy systems. The standard is called Fuzzy Markup Language (FML) which defines a new W3C XML-based language for fuzzy inference systems. In order to facilitate the design of FML-based fuzzy inference systems, an open-source Java library for FML, named JFML, is distributed [2] [3]. In this paper, we use JFML to design a fuzzy inference system for FML-based machine learning competition for Human and Smart Machine Co-Learning on Game of Go [4]. The goal of this competition is to design accurate and interpretable fuzzy rule-based inference systems using FML.

In this paper, we propose an evolutionary FML-based fuzzy inference system. The proposed method optimizes the rule base (RB) and the knowledge base (KB) using a genetic algorithm (GA). We use the mean square error (MSE) as an objective function in GA. In addition, we apply a feature selection method for obtaining more accurate systems. We apply our method to two tasks using the modified dataset [5]. Experimental results show that the proposed method can achieve more accurate inference performance by selecting some features from the original data.

This paper is organized as follows. Section II explains the proposed method. Section III shows computational experiments. Concluding remarks are presented in Section IV.

II. Evolutionary FML-based Fuzzy Inference Systems

Our evolutionary FML-based fuzzy inference system is composed of two optimization phases: the RB optimization and the KB optimization. Feature selection is also applied in our method. In this section, we present the general framework of the proposed method. Moreover, the procedure of the RB optimization, the KB optimization, and feature selection are presented in detail.

A. General Framework

In the general framework, the RB optimization and the KB optimization phases are repeatedly performed until the pre-defined stopping criterion is satisfied. Algorithm 1 shows a brief procedure of the general framework.

Algorithm 1 : General Framework

Require: D_{tra} (training dataset),
KB_{best} (the best knowledge base), RB_{best} (the best rule base)
Initialize(KB_{best})
D_{eva} = *Sampling*(D_{tra})
$RB_{best} = \phi$
while the stopping criterion is not met **do**
  RB_{best} = *RuleBase_Opt*(KB_{best}, RB_{best}, D_{tra}, D_{eva})
  KB_{best} = *KnowledgeBase_Opt*(KB_{best}, RB_{best}, D_{tra}, D_{eva})
end while
return KB_{best}, RB_{best}

1) Knowledge Base Initialization: First, the KB is defined with given fuzzy sets. Table I shows initial fuzzy sets where c and σ are the mean and the variance in Gaussian distribution, respectively. Fig. 1 shows the shapes of fuzzy sets. These fuzzy sets are assigned to all of the input variables as linguistic labels.

TABLE I. Parameters of Initial Fuzzy Sets.

Name	Type	Parameter
Very Small	Gaussian Shape	$c = 0$, $\sigma = 0.105$
Small	Gaussian Shape	$c = 0.25$, $\sigma = 0.105$
Medium	Gaussian Shape	$c = 0.5$, $\sigma = 0.105$
Large	Gaussian Shape	$c = 0.75$, $\sigma = 0.105$
Very Large	Gaussian Shape	$c = 1$, $\sigma = 0.105$
Don't Care	Rectangular Shape	$a = 0$, $b = 1$

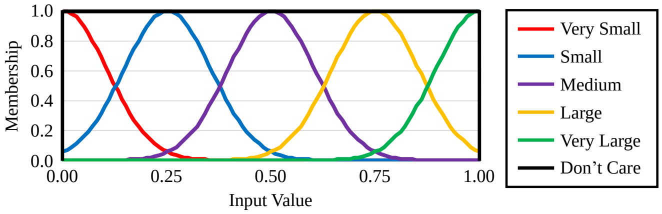

Fig. 1. Membership functions of the initial knowledge base.

2) Evaluation Dataset Preparation: The training dataset is divided into two subsets. One is used as the learning dataset for calculating the consequent part of each rule. The other is used as the evaluation dataset for evaluating an individual in GA. In this paper, the size of the evaluation dataset is specified as 500. 500 patterns are randomly selected from the trainig data by sampling without replacement and the learning dataset is composed of the remaining patterns.

B. Rule Base Optimization Phase

In the RB optimization phase, an individual is composed of a set of fuzzy if-then rules. Algorithm 2 shows the procedure of the RB optimization phase. In this phase, the given best knowledge base is used for calculating membership values.

Algorithm 2 : $RuleBase\_Opt(KB_{best}, RB_{best}, D_{tra}, D_{eva})$

Require: N_R (population size), N_{rule} (initial rule set size)

Initialize P_R by a heuristic rule generation method
$Evaluate(P_R, D_{tra}, D_{eva})$
while the stopping criterion is not met **do**
  $P_R{}' = Mating\_Selection(P_R)$
  $Mutation(P_R{}')$
  $Rule\_Pruning(P_R{}')$
  $Evaluate(P_R{}', D_{tra}, D_{eva})$
  $Q = P_R \cup P_R{}'$
  $P_R = Population\_Update(Q, N_R)$
end while
$RB_{best} = best(P_R)$
return RB_{best}

1) Population Initialization and Evaluation: A set of rules generated by a heuristic rule generation method. In the heuristic rule generation method, a fuzzy if-then rule is generated to cover the input space of a randomly selected pattern. The most compatible fuzzy set is used for each attribute. In initialization, each individual is composed of the number of rules N_{rule}. Thus, N_{rule} patterns are sampled in the heuristic rule generation method for each individual.

The individual is evaluated after defining the consequent part of each rule. The consequent part C_n of the nth rule is defined by Eq. (1):

$$C_n \leftarrow C_n + \eta\left(T_k - O_k\right)\frac{\mu_n\left(\mathbf{x}_k\right)}{\sum_{i=1}^{N_R}\mu_i\left(\mathbf{x}_k\right)}, \qquad (1)$$

where η is the learning rate, T_k is the desired output of kth learning pattern $\mathbf{x}_k$, O_k is the output predicted by the fuzzy system for the kth learning pattern, and $\mu_n(\cdot)$ is the membership value of the nth rule for a learning pattern. Finally, each individual is evaluated by calculating an MSE for the evaluation dataset D_{eva}.

2) Optimization Process for RB: RB is optimized by GA. To obtain the optimal rule base RB_{best}, we perform the following steps until a stopping criterion is satisfied:

Step 1) ***Mating Selection***: Select two parents using the Binary Tournament Selection. Then, a new individual inherits the rules at random from two selected parents.

Step 2) ***Mutation***: According to the mutation rate for each rule, randomly replace one fuzzy set (including *"don't care"*) of the antecedent part with another fuzzy set or *"don't care"*.

Step 3) ***Rule Pruning***: The rule prescreening is significantly effective for GA-based rule selection [6]. If there exists a rule whose membership values of all the attributes for all learning patterns are equal to or less than 0.5, replace the rule with a new rule generated by the heuristic rule generation method. At this time, if there exists a learning pattern whose membership value of even one attribute is equal to or less than 0.5 in all the rules, the pattern is primarily used for the heuristic rule generation method.

Step 4) ***Evaluation of the Offspring***: Evaluate individuals in the offspring population in the same manner as the evaluation of the initial population.

Step 5) ***Population Update***: Select N_R individuals which have a lower fitness value (i.e., lower error rate) in the current and offspring populations in order to make the new population in the next generation.

Step 6) ***Termination Condition***: If the termination condition is satisfied, output the optimal rule set of the best individual in the final generation.

C. Knowledge Base Optimization Phase

In the KB optimization phase, an individual is composed of a series of parameters defining the shapes of membership functions. Algorithm 3 shows the procedure of KB optimization phase.

Algorithm 3 : $KnowledgeBase\_Opt(KB_{best}, RB_{best}, D_{tra}, D_{eva})$

Require: N_K (population size)

Initialize P_K by $Perturbation(KB_{best})$
$Evaluate(P_K, D_{tra}, D_{eva})$
while the stopping criterion is not met **do**
  $P_K{}' = Mating\_Selection(P_K)$
  $Perturbation(P_K{}')$
  $Evaluate(P_K{}', D_{tra}, D_{eva})$
  $Q = P_K \cup P_K{}'$
  $P_K = Population\_Update(Q, N_K)$
end while
$KB_{best} = best(P_K)$
return KB_{best}

1) Population Initialization and Evaluation: First, a set of parameters in the KB_{best} is added to the initial population. Remaining $(N_k - 1)$ individuals are initialized by perturbing parameters in the KB_{best}. The evaluation is the same manner as the evaluation in the RB optimization phase.

2) Optimization Process for KB: In this paper, each fuzzy set is represented by the two parameters (i.e., the mean and the variance) since we use gaussian shape as a fuzzy set. Thus, the length of genes per individual is always multiple of two. In order to remain the shape, the crossover point is always set to be a number of multiple of two.

To obtain the optimal set of parameters of KB_{best}, we perform the following steps until a stopping criterion is satisfied:

Step 1) ***Mating Selection***: Select two parents using the Binary Tournament Selection. Then, a new individual is generated by single point crossover from parents.
Step 2) ***Perturbation***: According to the perturbation rate for each gene, add a random real number. However, for the mean parameter, if a new value over the mean of the neighbor fuzzy set, the perturbation is canceled to prevent conflict.
Step 3) ***Evaluation of the Offspring***: Generated offspring are evaluated in the same manner as the RB optimization phase.
Step 4) ***Population Update***: Select N_K individuals which have a lower fitness value from the current population and offspring. Then, set them as the population in the next generation.
Step 5) ***Termination Condition***: If the termination condition is satisfied, output the optimal KB of the best individual from the population in the final generation.

D. Feature Selection Method

We apply a sequential forward feature selection method [7] to our proposed general framework. This is a wrapper method, in which an effective feature is incrementally added through iterative experiments with our proposed general framework.

First, before an execution of the feature selection method, we randomly select a validation dataset D_{valid} from the given training dataset D. In this paper, the size of D_{valid} is set to 1/3 of that of D. The size is almost as many as that of the test dataset D_{test}, which is provided in this competition. The remaining dataset is used for the training dataset D_{tra} in Algorithm 1.

Next, we start feature selection from feature arrays which include one feature. If the number of features is three, feature arrays {1, 0, 0}, {0, 1, 0}, and {0, 0, 1} is prepared and each execution of the proposed general framework using each feature array is performed. After that, each designed fuzzy inference system using each feature array is evaluated by MSE value for D_{valid}. Then, we select a feature array which is used by the fuzzy inference system with the minimum MSE value for D_{valid}. Finally, the other feature is added into the selected feature array (i.e., new feature arrays are {1, 1, 0} and {1, 0, 1} if {1, 0, 0} is selected) and we restart each execution of the proposed general framework using each new feature array. In this way, we obtain the order of adding effective features.

After the execution of a feature selection method, the order for adding effective features is obtained. We use the order to get an optimal combination of selected features. Each combination of selected features based on the obtained order is evaluated for an MSE value for D_{valid}. We finally obtain a fuzzy inference system with the minimum MSE value for D_{valid} as a system using an optimal combination of selected features. D_{valid} is used for the evaluation of the generalization ability of a fuzzy inference system. In this paper, a combination of selected features is represented by an array of binary values. Each value is assigned to each feature. The *i*-th value is set to 1 if *i*-th feature is used for a fuzzy inference system. In the same way, the *i*-th value is set to 0 if *i*-th feature is not used.

III. Computational Experiments

A. Dataset Preprocessing

The given datasets are composed of a black player's move and a white player's move. Thus, the given datasets include missing values in some patterns if a game ends on a black player's turn. However, the proposed method does not consider patterns including such missing values. Thus, for the proposed method, we throw away patterns including missing values from the given datasets in advance. We also normalize attribute values by the maximum and minimum values of all the variables into the range of [0, 1].

B. Experimental Settings

The learning rate η is set to 0.5, the epoch of learning is set to 100, the crossover rate for two phases is set to 0.9, the mutation rate for the RB optimization phase is set to 0.1, the perturbation rate for the KB optimization phase is set to 0.8, the size of the evaluation dataset D_{eva} is set to 500, and the number of rules per rule set is set to be in the range of [500, 5,000]. The number of generations for both phases is set to 20, the population size in both phases is set to 20, and the number of iterations for the general framework is set to 3.

C. Experimental Results

Table II shows that MSEs for D and D_{test} by obtained fuzzy inference systems using the finally selected features. We show experimental results for DBWR(t+1) and EBWR(t) in Tables III and IV, respectively. These tables show the selected effective features, MSE for D, MSE for the test dataset, and MSE for D_{valid}. The best results for the same dataset are shown in red. We finally obtain a fuzzy inference system using selected features by the minimum MSE for D_{valid}. In Tables III and IV, the finally selected feature array is shown in red.

TABLE II. MSEs of the Obtained Fuzzy Inference System.

MSE for DBWR(t+1)		MSE for EBWR(t)	
Training	**Test**	**Training**	**Test**
8.77e-04	5.76e-04	4.71e-02	1.18e-01

TABLE III. Experimental Results for DBWR(t+1).

Feature Array	MSE		
	D	Test	D_{valid}
000000000100	1.15e-03	7.99e-04	1.21e-03
000100000100	9.49e-04	6.17e-04	1.02e-03
000100001100	**8.77e-04**	**5.76e-04**	**9.71e-04**
010100001100	9.05e-04	6.64e-04	1.05e-03
010110001100	8.88e-04	6.39e-04	9.97e-04
010110011100	9.15e-04	6.50e-04	1.04e-03
010110011110	8.86e-04	6.25e-04	9.95e-04
010110011111	9.40e-04	6.80e-04	1.07e-03
011110011111	9.04e-04	6.25e-04	1.00e-03
011110111111	9.47e-04	7.03e-04	1.07e-03
011111111111	9.33e-04	6.55e-04	1.04e-03
111111111111	9.81e-04	7.12e-04	1.13e-03

TABLE IV. EXPERIMENTAL RESULTS FOR EBWR(t).

Feature Array	MSE		
	D	**Test**	**D_{valid}**
100000000000	1.77e-01	1.82e-01	1.73e-01
100000000100	6.52e-02	**9.16e-02**	6.50e-02
100000000101	5.55e-02	1.10e-01	5.77e-02
100000000111	4.79e-02	1.03e-01	5.08e-02
101000000111	4.67e-02	1.08e-01	4.93e-02
111000000111	4.86e-02	1.29e-01	5.17e-02
111000100111	**4.62e-02**	1.18e-01	5.01e-02
111100100111	4.66e-02	1.19e-01	5.08e-02
111100110111	4.76e-02	1.15e-01	5.08e-02
111101110111	4.74e-02	1.27e-01	5.00e-02
111111110111	4.71e-02	1.18e-01	**4.89e-02**
111111111111	4.84e-02	1.07e-01	4.90e-02

We show the predicted win rate curves for DBWR(t+1) in Figs. 2 and 3. We also show the predicted win rate curves for EBWR(t) in Figs. 4 and 5. These win rate curves are predicted by fuzzy inference systems using the selected features by the minimum MSE for D_{valid}.

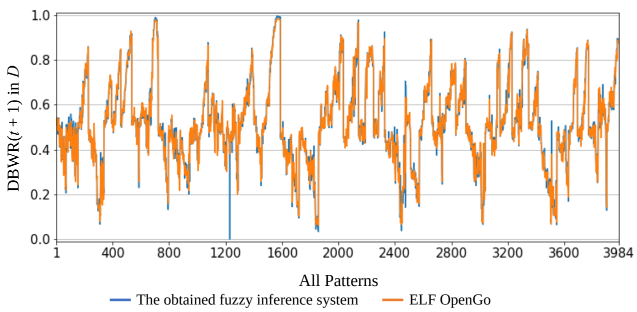

Fig. 2. Predicted DBWR(t+1) curves for the training dataset D.

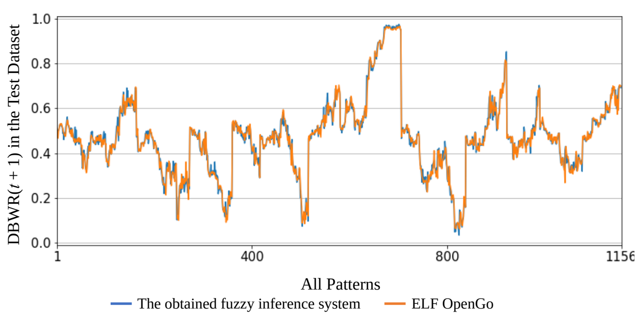

Fig. 3. Predicted DBWR(t+1) curves for the test dataset.

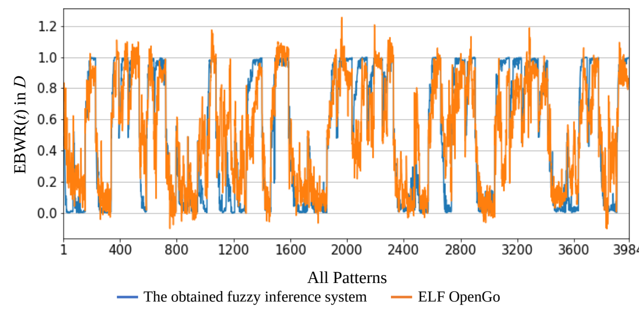

Fig. 4. Predicted EBWR(t) curves for the training dataset D.

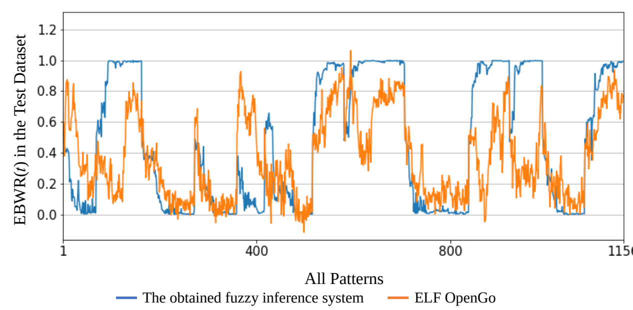

Fig. 5. Predicted EBWR(t) curves for the test dataset.

IV. CONCLUSION

We proposed an evolutionary FML-based fuzzy inference system with a feature selection method. The computational experiments showed that the proposed method can efficiently generate a fuzzy inference system with good prediction performance for DBWR(t+1).

As a future research topic, we consider a better way to select a feature array for minimizing MSE for the test dataset.

REFERENCES

[1] *IEEE Standard for Fuzzy Markup Language*, Standard 1855-2016, 2016, pp. 1-89, Available: https://standards.ieee.org/findstds/standard/1855-2016.html.

[2] Java Fuzzy Markup Language (https://www.uco.es/JFML/)

[3] J. M. Soto-Hidalgo, Jose M. Alonso, G. Acampora, and J. Alcala-Fdez, "JFML: A java library to design fuzzy logic systems according to the IEEE Std 1855-2016," *IEEE Access*, vol. 6, pp. 54952-54964, 2018.

[4] FML-based Machine Learning Competition for Human and Smart Machine Co-Learning on Game of Go at IEEE WCCI 2020 (http://oase.nutn.edu.tw/wcci2020-fmlcompetition/)

[5] FML Competition in FUZZ-IEEE 2020 (https://github.com/CI-labo-OPU/FML_Competition2020)

[6] H. Ishibuchi, T. Nakashima, M. Nii, *Classification and Modeling with Linguistic Information Granules*, Springer-Verlag Berlin, Heidelberg, 2004, pp. 185-187.

[7] P. Pudil, J. Novovičová, and J. Kittler, "Floating search methods in feature selection," *Pattern Recognition Letters*, vol. 15, no. 11, pp. 1119-1125, November 1994.